Improved RRS logical architecture using genetic algorithm

Shim, Hyo-sub / Jung, Jae-chun
KINGS, 658-91, Haemaji-ro, Seosaeng-myeon, Ulju-gun, Ulsan, Korea
Hyosub1979@daum.net

1. Introduction

An improved RRS (Reactor Regulating System) logic is implemented in this work using systems engineering approach along with GA (Genetic Algorithm) deemed as providing an optimal solution to a given system. The current system works desirably and has been contributed to the safe and stable NPP (Nuclear Power Plant) operation. However, during the ascent and decent section of the reactor power, the RRS output reveals a relatively high steady state error and the output also carries a considerable level of overshoot. In an attempt to consolidate conservatism and minimize the error, this research proposes applying genetic algorithm to RRS and suggests reconfiguring the system. Prior to the use of GA, reverse-engineering is implemented to build a Simulink-based RRS model and re-engineering is followed to apply the GA and to produce a newly-configured RRS generating an output that has a reduced steady state error and diminished overshoot level. A full-scope APR1400 simulator is used to observe and extract the inputs and output of RRS.

2. Methods and results

Reverse-engineering was exploited to identify the due constraints of this research by reviewing the requirements of a source document [1] and subsequently to construct a Simulink-based executable RRS logic architecture (SRRS) by retrieving the input and output (CES, Compensated Error Signal) data from the simulator. As shown in Fig. 1, lead-lag compensator (LLC), HPF (High Pass Filter) and lag compensator constitutes RRS and its output is CES. A comparison between the simulator CES and the SRRS CES was conducted and the result demonstrated a discrepancy. To cope with the discrepancy, the parameters of the two compensators and one filter were manually tweaked.

Following the modeling of the current RRS, reengineering was performed 1) to devise the conceptual design of the improved RRS (IRRS) and 2) to build IRRS using an existing GA module which were in turn tailor-made for this research. At the end, the theoretical and empirical verifications for IRRS were executed in order to substantiate the stability of IRRS.

2.1 Acquire simulator input and CES data

The acquisition of the data was aimed at emulating RRS in Simulink which enables to build a graphical executable model to simulate it and obtain a multitude of experimental results. The data include Tavg [°C], Tref [°C], Rx_PWR (Reactor Power, %) and TLI (%). All of the data acquisitions are available in the fullscope APR 1400 simulator. The subtraction of Tavg and Tref was defined as Input_1 and the one of Rx_PWR and TLI was named Input_2 respectively.

2.2 Modeling and simulation

This section was committed to developing SRRS working in the same way as the simulators RRS. Following the acquisition of the simulator data vital to build SRRS, the initial version of SRRS was modeled and simulated

2.3 Synchronize CES of simulator and SRRS

A discrepancy occurred as depicted in Fig. 2 between the simulator CES and SRRS CES and it was ascribed to the different platforms of the simulator and SRRS. The simulator is built on 3KEYMASTER, simulation software and SRRS was modeled in Simulink. To resolve this issue, the parameters of LLC, HPF and lag compensator were adjusted and at each time, the scope output was observed. The process of the adjustment and observation were executed continuously one another until the discrepancy remained infinitesimal

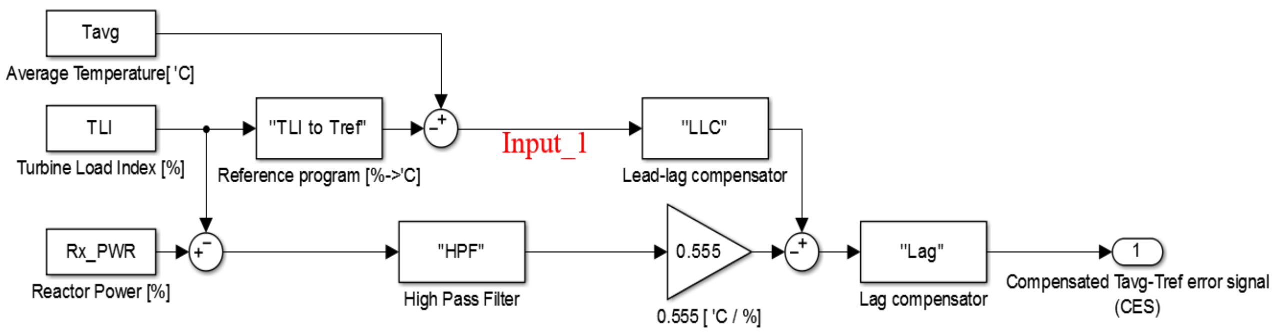

Fig. 1. RRS Schematic Diagram

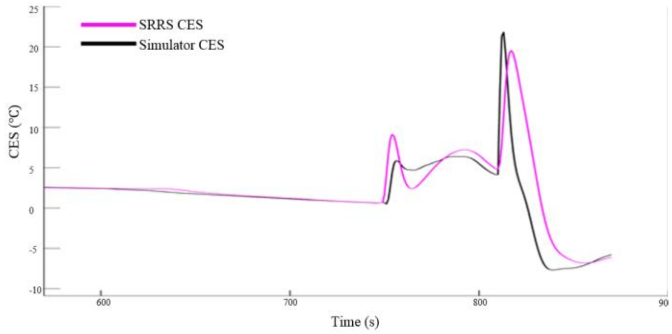

Fig. 2. CES discrepancy between simulator and SRRS

The discrepancy was fixed by fine-tuning the mathematical expressions of the compensators and filter. The fixed SRRS was renamed as MRRS (Modified RRS).

2.4 Formulate improved RRS logic architecture

The primary objective of this research was to design IRRS whose CES was meant to finely follow input_1. The simulator affords multifarious scenarios, some of which indicated that RRS improvements would be made. They concerned 25%, 50%, 75%, 100% power level scenarios. Thus, IRRS was proposed to enhance 1) the initial CES at 25% power level, 2) input_1-CES deviation at 50-100% power level and 3) overshoot in a transient. A conceptual additional LLC was integrated with MRRS and it was positioned at the end of the lag compensator. Soon an IRRS Simulink model was formulated by adding the conceptual ALLC to MRRS. The mathematical expression of ALLC was generated in Section 2.5.

2.5 Construct SGA for development of additional LLC

An existing SGA module introduced in a technical book [10] was reviewed to devise ALLCs mathematical expression using the module. The Labview-oriented module was carefully studied and modeled in Labview to comprehend the principal of SGA. The module, which consists of several sub-modules and has various inputs, was partly modified for this paper and was, renamed MSGA. The iterated execution happed to establish the effective ranges of the parameters. In the final stage, ALLCs mathematical expression was derived and verified. The verified ALLC was deployed to the IRRS Simulink model.

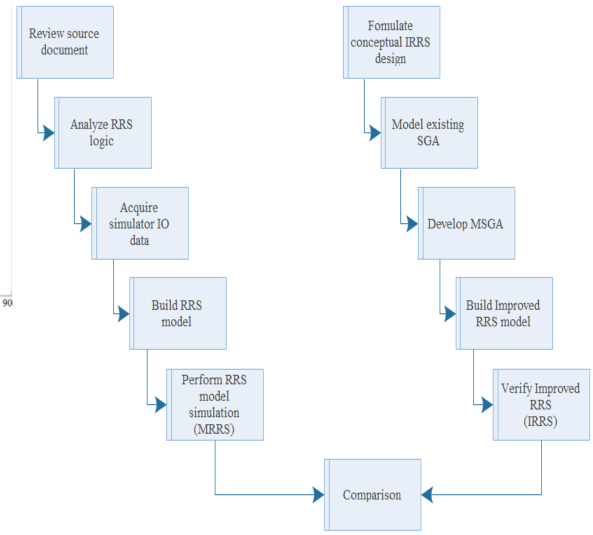

Fig. 3. Overall scheme for GA-based RRS logic architecture improvement

2.6 Derive TTF (Total Transfer Function) of IRRS

The TTF of IRRS was computed by using the Matlab linear analysis as well as by reshaping the IRRS Simulink model base on equivalent block diagram [13]. Two linear analysis points (Open-loop input) were applied at the end of input_1 and input_2 and at the end of ALLC, one linear analysis point (Open-loop output) was also set. In response, the linear analysis tool was executed and the step plot function of the tool was chosen. The tool created a file which was in turn transferred to the Matlab workspace. Later, the IRRS model was reshaped using equivalent block diagrams and accordingly, the Input_1, LLC, 1/HPF, HPF, lag compensator and ALLC were connected in series. Each of the mathematical expressions was written in the Matlab editor and their multiplication was run to derive the TTF of IRRS

2.7 TTF comparison and selection

The IRRS TTF was intended to determine the stability of IRRS as the empirical verification is only valid after the stability is validated [12]. The TTF computed by the linear analysis was named TTF-1 and the one developed by the equivalent block diagram was dubbed TTF-2. TTF-1 and TTF-2 were compared to be singled out for the theoretical verification. It was observed that the TTF-2s fitness to the IRRS CES outpaced the one of TTF-1 and was consequently selected for the stability verification.

2.8 Theoretical verification with selected TTF

Three methods for the verification included Routh Hurwitz stability criterion, Root locus and bode diagram and they were introduced to ascertain the IRRS stability with TTF-2. TTF-2 was inputted in each method and each consequence resulted in substantiating the IRRS stability.

2.9 Empirical verification with simulator

Following the theoretical verification, the empirical verification was staged by generating a scope representing Input_1, IRRS CES and MRRS CES. The generation featured the same scenarios in Section 2.4. This result revealed that the highest IRRS CES during the initial stage of the 25% power level was smaller than the MRRS CES and the enhancement resulted in consolidating the conservatism of RRS and in further deterring the potential uncontrolled motion of CEA at low power levels. The 50%, 75% and 100% power levels of IRRS were also observed. Smaller CESs and lower overshoots were monitored at each level.

3. Conclusion

This paper was designed for RRS to generate a stable and optimal CES which is delivered to DRCS to automatically and rapidly regulate CEA motions. For the purpose of the stable and optimal CES, the overshoot and accuracy held the dominant position in this work and led to devising IRRS. The performances of IRRS were analyses in the result section and verified with normal power operations at multiple power levels and a transient condition initiated by the sequential trips of two FWPs. On the sidelines of this objective of this paper, the oscillation observed in MRRS was smoothed out by IRRS. The average MRRS input_1-CES deviation was indicated at 0.17923°C. The measurement was executed once again with IRRS. The average IRRS input_1-CES deviation was gauged at 0.033352°C, down 0.145881°C from 0.17923°C.

4. Acknowledgement

This research was supported by the 2016 Research Fund of the KEPCO International Nuclear Graduate School (KINGS), Republic of Korea.

REFERENCES

[1] United States Nuclear Regulatory Commission, APR1400 Design Control Document, 2014. Available from: http://www.nrc.gov [cited 2016 May 20]
[2] Kim Sangjin, Song Byeonggeun, Oh Sejun, Yu Samsang, Advanced Automatic Control, 1st ed. ISBN 978-89-5526-832-4
[3] K. KRISHNAKUMAR, DAVID E. GOLDBERG, Control System Optimization using Genetic Algorithms, Journal of Guidance, Control, and Dynamics 15 (1992) 735-740.
[4] Reis, L., Porto, R., and Chaudhry, F, Optimal Location of Control Valves in Pipe Networks by Genetic Algorithm. J. Water Resour. Plann. Manage. 123 (1997) 317-326.
[5] Cheng-Jian Lin, A GA-based Neural Fuzzy System for Temperature Control. Fuzzy Sets and Systems. 143 (2004) 311-333.
[6] S.P. Ghoshal, Application of GA/GA-SA based Fuzzy Automatic Generation Control of a Multi-Area Thermal Generating System. Electric Power Systems Research. 70 (2004) 115-127.
[7] Tzuu-Hseng S. Li, Ming-Yuan Shieh, Design of a GA-based Fuzzy PID Controller for Non-Minimum Phase Systems. Fuzzy Sets and Systems. 111 (2000) 183-197.
[8] P.J. FLEMING, R.C. PURSHOUSE, Genetic Algorithms in Control Systems. Department of Automatic Control and Systems Engineering, University of Sheffield, UK, 2001.
[9] Western Services Corporation, 3KEYMASTER. Available from: http://www.power-technology.com [Cited 2016 June 10]
[10] Kim Hyoil, Jin Ganggyu, Jeon Seunghwan, Genetic algorithm learning on Labview, 1st ed. ISBN 978-89-92649-80-3
[11] National Instruments, Labview. Available from: http//www.ni.com [Cited 2016 June 12]
[12] Kim Seul, Control System Analysis and Simulink applications, 2nd ed. ISBN 978-89-6364-063-1
[13] WIKIBOOKS, Control Systems / Block Diagrams. Available from: https://en.wikibooks.org [Cited 2016 June 15]
[14] WIKIPEDIA, Root locus. Available from: https://en.wikipedia.org [Cited 2016 June 18]
[15] Choi Yeonuk, Simulink comprehension and actual controller design, 1st ed. ISBN 979-11-5600-386-1